On Precision - Redundancy Relation in the Design of Source Coding Algorithms

Yuriy A. Reznik

Qualcomm Incorporated

5775 Morehouse Drive, San Diego, CA 92122

E-mail: `yreznik@ieee.org`

Abstract

We study the effects of finite-precision representation of source's probabilities on the efficiency of classic source coding algorithms, such as Shannon, Gilbert-Moore, or arithmetic codes. In particular, we establish the following simple connection between the redundancy R and the number of bits W necessary for representation of source's probabilities in computer's memory (R is assumed to be small):

$$W \lesssim \eta \log_2 \frac{m}{R} \,,$$

where m is the cardinality of the source's alphabet, and $\eta \leqslant 1$ is an implementation-specific constant. In case of binary alphabets ($m = 2$) we show that there exist codes for which $\eta = 1/2$, and in m-ary case ($m > 2$) we show that there exist codes for which $\eta = m/(m+1)$. In general case, however (which includes designs relying on progressive updates of frequency counters), we show that $\eta = 1$. Usefulness of these results for practical designs of source coding algorithms is also discussed.

1 Introduction

Since Shannon it is known that the average rate of a code constructed for a stochastic source cannot be lower than the entropy of this source. The difference between the rate and the entropy is called the *redundancy* of the code.

It is also known, that the redundancy of a code fundamentally depends on the number of input symbols that are jointly mapped into a code during the encoding process. This number (say, n) is usually called a *block-size* or *delay* of the code. The ratio of redundancy over n is called the *redundancy rate* of a code, and the speed of its convergence has long been considered a key criterion in understanding the effectiveness of source codes.

For example, it was shown that many classic codes for known memoryless sources (such as block Shannon, Huffman, or Gilbert-Moore codes [4, 9]) attain the redundancy rate of $R = O(1/n)$ [6, 2, 9, 15, 16]. Krichevsky-Trofimov codes for a class of memoryless sources achieve the rate of $O(\log n/n)$ [9]. Lempel-Ziv codes for this class were shown to converge at the rate of $O(\log \log n/ \log n)$ [19, 18], etc.

At the same time, much less is known about the connection between the redundancy and typical implementation constraints, such as width of computer's registers.

Perhaps the best example of a prior effort in exploring this connection was the development of finite-precision implementations of arithmetic codes (cf. Pasco [10], Rubin [13], Witten et al. [17]). Simple redundancy bounds for such finite-precision algorithms were offered by Han and Kobayashi [8] and Ryabko and Fionov [14]. However, both of these results are specific to particular implementations.

In this paper, we pose and study somewhat more general questions:

- *"What is the achievable redundancy of a code that can be constructed by a machine with W-bits representations of source's probabilities?"*, and/or

- *"What is the smallest number of bits that one should use for representing source's probabilities in order to construct a code with target redundancy R?"*

For simplicity, we assume that codes can use infinite delays, and that the redundancy is caused only by errors of finite precision representations of probabilities.

Indeed, if the answer to the first question is known, then it can be treated as a lower bound for achievable redundancy of any source coding algorithm with such constraint on internal representation of source's probabilities (or their estimates). The answer to the second question is of immediate practical interest.

This paper is organized as follows. In Section 2, we present formal setting of our problems and describe our results. Their possible applications are discussed in Section 3. All proofs and supplemental information from Diophantine approximation theory (which we use in order to derive our results) are given in Appendix A.

2 Definitions and Main Results

Consider an m-ary memoryless source S, producing symbols $\alpha_1, \ldots, \alpha_m$ with probabilities $p_1, \ldots, p_m$. We assume that in our source coding algorithm instead of true probabilities $p_1, \ldots, p_m$, we have to use their approximations: $\hat{p}_1, \ldots, \hat{p}_m$.

By $\delta_i = p_i - \hat{p}_i$ $(i = 1, \ldots, m)$ we denote probability differences (errors) for each symbol, and by

$$\delta^* = \max_i |p_i - \hat{p}_i| = \max_i |\delta_i| , \tag{1}$$

we denote their maximum absolute value. Further, by $p_{\min}$ we denote the smallest probability in the source:

$$p_{\min} = \min_i p_i \tag{2}$$

and we will assume that $p_{\min} > 0$, and that it is also relatively large compared to our maximum approximation error:

$$\frac{\delta^*}{p_{\min}} < 1 . \tag{3}$$

2.1 Upper Bounds for Redundancy

We first show that, under above described conditions, the loss in compression efficiency incurred by using $\hat{p}_1, \ldots, \hat{p}_m$ is bounded by the following simple expression.

Lemma 1.

$$D\left(p \,||\, \hat{p}\right) \leqslant m\delta^* \frac{1}{1 - \delta^*/p_{\min}} = m\delta^* \left(1 + O\left(\delta^*\right)\right) . \tag{4}$$

We next consider a more specific source coding scheme, in which source's probabilities are approximated by rational values:

$$\hat{p}_i = \frac{f_i}{t}, \quad f_i, t \in \mathbb{N} \quad (i = 1, \ldots, m) \tag{5}$$

where

$$t = \sum_{i=1}^{m} f_i \tag{6}$$

is the common denominator.

We derive two results regarding attainable redundancy due to such approximations. We start with a more general (and weaker) statement:

Theorem 1. *Given any $t \gg 1$, it is possible to find rational approximations of source's probabilities (5), such that:*

$$D(p \| f/t) \leqslant \frac{m}{2t} \frac{1}{1 - 1/(2t\, p_{\min})} = \frac{m}{2t} \left(1 + O\left(t^{-1}\right)\right). \tag{7}$$

This shows that redundancy is decreasing approximately inverse proportionally to t. This bound is correct for any values t.

Nevertheless, a more detailed study of this problem (which involves the use of tools and results from Diophantine approximation theory [3]) reveals that among various possible values of denominator t, there exists some, for which precision of approximation (5) can be much higher. In turn, this leads to much lower redundancy of source codes based on such approximations.

We claim the following:

Theorem 2. *There exist infinitely many integers $t, f_1, \ldots, f_m$ $(m > 2)$ producing approximations of source's probabilities (5), such that:*

$$D(p \| f/t) < \frac{m}{t^{1+1/m}} \frac{1}{1 - 1/\left(t^{1+1/m}\, p_{\min}\right)} = \frac{m}{t^{1+1/m}} \left(1 + O\left(t^{-1-1/m}\right)\right). \tag{8}$$

In a case when $m = 2$, there exist infinitely many integers t, f_1 (with $f_2 = t - f_1$) producing approximations of source's probabilities (5), such that:

$$D(p \| f/t) < \frac{2\kappa}{t^2} \frac{1}{1 - \kappa/(t^2\, p_{\min})} = \frac{2\kappa}{t^2} \left(1 + O\left(t^{-2}\right)\right), \tag{9}$$

where:

$$\kappa = \begin{bmatrix} 5^{-1/2}, & \text{if } p_1 = \frac{r\psi+s}{u\psi+v}, \ \psi = \frac{\sqrt{5}-1}{2}, \ rv - us = \pm 1, \ r, s, u, v \in \mathbb{Z}, \\ 2^{-3/2}, & \text{otherwise}. \end{bmatrix} \tag{10}$$

The above result is not immediately obvious, as it implies that codes relying on rational approximations (5) and constructed for binary sources, can be much more precise than codes using equally large parameter t but constructed for m-ary sources $(m > 2)$. In general, based on the above result, the larger is the cardinality of the alphabet, the more severe is the effect of approximations of source's probabilities.

Recall, that most traditional redundancy bounds for codes for memoryless sources (such as delay-redundancy relations, obtained assuming infinite precision implementation) don't change their order based on cardinality of the alphabet. Our finding above indicates that for many practical algorithms this may no longer be a case once one start accounting redundancy contributions caused by finite-precision implementation.

2.2 Precision (and Memory Usage) vs Redundancy

Consider an implementation of a source coding algorithm employing rational approximations of source's probabilities (5). We assume that such an algorithm can either store immediate values f_i, or their cumulative equivalents (as it is convenient, e.g. in the design of Shannon, Gilbert-Moore, or arithmetic codes):

$$s_i = \sum_{\substack{j = 1, \ldots, i \\ \{k_j\} : p_{k_1} \leqslant \ldots \leqslant p_{k_m}}} f_{k_j} . \tag{11}$$

In either case, the maximum integer value that might be stored per symbol is t. Hence, the required *register width* or *memory cost per symbol W* in such a scheme is simply:

$$W = \lceil \log_2 t \rceil. \tag{12}$$

Using our prior results, we derive the following simple bounds for this quantity:

Corollary 1. *In the design of a source coding scheme, the number of bits W used for representing source's probabilities, and the resulting increase in redundancy R satisfy:*

$$W < \log_2 \left(\frac{m}{R} + \frac{1}{p_{\min}} \right) = \log_2 \left(\frac{m}{R} \right) + O(R) \tag{13}$$

Corollary 2. *There exist infinitely many implementations of source codes, in which the number of bits W used for representing source's probabilities and the resulting increase in redundancy R satisfy:*

$$W < \frac{m}{m+1} \log_2 \left(\frac{m}{R} + \frac{1}{p_{\min}} \right) + 1 = \frac{m}{m+1} \log_2 \left(\frac{m}{R} \right) + 1 + O(R), \tag{14}$$

where $m > 2$ is a cardinality of source's alphabet.

In a case when $m = 2$, there exist infinitely many implementations of binary source codes, in which the number of bits W used for representing source's probabilities and the resulting increase in redundancy R satisfy:

$$W < \frac{1}{2} \log_2 \left(\frac{2}{R} + \frac{1}{p_{\min}} \right) + \frac{1}{2} \log_2 4\kappa = \frac{1}{2} \log_2 \left(\frac{8\kappa}{R} \right) + O(R), \tag{15}$$

where κ is a constant defined in (10).

Here, it can be seen that the use of binary alphabets can lead up to a factor of 2 savings in the number of bits needed in representation of probabilities.

It should be noted, however, that this factor of 2 does not implies that conversion to binary alphabets will produce equivalent saving in overall storage usage. The problem here is that in order to implement such codes one need to store not only values f_1 or s_1, but also f_2 or s_2 or t. For example, if one would substitute an m-ary source with a cascade (binary tree) of $m - 1$ binary sources, then this would increase memory usage by a factor of $2(m-1)/m$, prior to a possibility to use gains predicted by our Corollary 2. Nevertheless, such a conversion can still be justified by the need to use shorter registers.

4

In general, given an m-ary source ($m \geqslant 2$), we can say that the amount of memory M that needs to be use for construction of its code (that involves rational representation of its probabilities (5)) is at least

$$M = mW \,, \qquad (16)$$

where W is connected to redundancy as predicted by our Corollaries 1 and 2.

3   Conclusions

Several bounds establishing connection between redundancy and precision of representation of probabilities in source coding algorithms have been derived.

These results can be used for maximizing performance of codes with respect to precision or memory available on each particular computing platform.

The results of our Theorem 2 (and Corollary 2) revealing the existence of higher-precision approximations, particularly for sources with small cardinality of alphabets may also influence future practical designs of source coding algorithms. For example, in handling of sources with large alphabets one may consider grouping of their symbols and using a cascade of codes with smaller alphabets as a way to improve their performance given register-width constraints.

References

[1] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, Dover, NewYork, 1972.

[2] R. Capocelli and A. de Santis, Tight Upper bounds on the Redundancy of Huffman Codes, IEEE Trans. Information Theory, 35, pp. 1084–1091, 1989.

[3] J. W. S. Cassels, *An Introduction to Diophantine Approximations*, Cambridge University Press, 1957.

[4] T. M. Cover and J. M. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[5] M. Drmota, H-K. Hwang, and W. Szpankowski, Precise Average Redundancy of an Idealized Arithmetic Coding, Proc. Data Compression Conference, pp. 222-231, Snowbird, UT, 2002.

[6] R. Gallager Variations on the Theme by Huffman, IEEE Trans. Information Theory, 24, pp. 668–674, 1978.

[7] M. Gröetschel, L. Lovácz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, Springer, Berlin, 1988.

[8] T. S. Han, and K. Kobayashi, *Mathematics of Information and Coding*, American Mathematical Society, Boston, MA, 2001.

[9] R. E. Krichevsky, *Universal Data Compression and Retrieval*, Kluwer, Norwell, MA, 1993.

[10] R. Pasco, Source coding algorithm for fast data compression, Ph.D. thesis, Stanford University, 1976.

[11] J. J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM J. Res. Dev.*, vol. 20, pp. 198–203, 1976.

[12] J. J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM J. Res. Dev.*, vol. 23, no. 2, pp. 149–162, 1979.

[13] F. Rubin, "Arithmetic stream coding using fixed precision registers," *IEEE Trans. Inform. Theory*, vol. 25, no. 6, pp. 672–675, 1979.

[14] B. Ya. Ryabko and A. N. Fionov, Efficient Method of Adaptive Arithmetic Coding for Sources with Large Alphabets, *Probl. Inform. Transm.*, vol. 35, pp. 95–108, 1999. (in Russian).

[15] P. Stubley, On the redundancy of optimum fixed-to-variable length codes, *Proc. Data Compression Conference*, pp. 90-97, Snowbird, UT, 1994.

[16] W. Szpankowski, Asymptotic Average Redundancy of Huffman (and Other) Block Codes, *IEEE Trans. Information Theory*, vol. 46, no. 7, pp. 2434–2443, 2000.

[17] I. H. Witten, R. Neal, J.G. Cleary, Arithmetic Coding for Data Compression, *Comm. ACM.*, vol. 30, no. 6, pp. 520–541, 1987.

[18] A.D. Wyner and J. Ziv, Fixed Data Base Version of the Lempel-Ziv Data Compression Algorithm, *IEEE Trans. Information Theory*, vol 37, pp. 878–880, 1991.

[19] J. Ziv and A. Lempel, A Universal Algorithm for Sequential Data Compression, *IEEE Trans. Information Theory*, vol. 23, 337–343, 1977.

A   Proofs

We first prove statement of our Lemma 1:

$$
\begin{aligned}
D\left(p\,\|\,\hat{p}\right) &= \sum_{i=1}^{m} p_i \log\left(\frac{p_i}{\hat{p}}\right) \\
&= -\sum_{i=1}^{m} p_i \log\left(1 - \delta_i/p_i\right) \\
&\leqslant -\sum_{i=1}^{m} p_i \log\left(1 - \delta^*/p_i\right) \\
&\leqslant \sum_{i=1}^{m} p_i \frac{\delta^*/p_i}{1 - \delta^*/p_i} \\
&\leqslant \sum_{i=1}^{m} \frac{\delta^*}{1 - \delta^*/p_{\min}} \\
&= m\delta^* \frac{1}{1 - \delta^*/p_{\min}}
\end{aligned}
\tag{17}
$$

where (17) is due to the following inequality [1]:

$$-\log(1-x) \leqslant \frac{x}{1-x} \qquad (x < 1).$$

We next consider rational approximations (5) of source probabilities. By assuming that factors f_i are chosen such that

$$\delta_i = |p_i - f_i/t| = \frac{1}{t}\,|t\,p_i - f_i| = \frac{1}{t}\,\min_{z \in \mathbb{Z}} |t\,p_i - z|\ ,$$

we can show that:

$$\delta_i \leqslant \frac{1}{2\,t}\ ,$$

and, consequently:

$$\delta^* \leqslant \frac{1}{2\,t}\ .$$

This, combined with Lemma 1 leads to a statement of our Theorem 1.

In order to prove first statement of Theorem 2, we will need to use the following result regarding attainable precision of simultaneous Diophantine approximations [3, p. 14, Theorem III]:

Fact 1. *For any $n > 1$ irrational values $\alpha_1, \ldots, \alpha_n$, there exist infinitely many sets of integers $(a_1, \ldots, a_n, q)$ such that:*

$$q^{1/n} \max\{|q\,\alpha_1 - a_1|, \ldots, |q\,\alpha_n - a_n|\} < \frac{n}{n+1}\ .$$

In our case, this means that there must exist rational approximations of set of probabilities $p_1, \ldots, p_m$, such that:

$$\delta^* < \frac{m}{m+1}\,t^{-1-1/m} < t^{-1-1/m}\ .$$

This, combined with Lemma 1, leads to bound (8) claimed by the Theorem 2.

Indeed, it shall be noted, that the task of finding such Diophantine approximations is not a trivial one, and the reader is referred to a book of M. Gröetschel, L. Lovácz, and A. Schrijver [7] which discusses this problem in details.

Consider now binary case ($m = 2$). We first notice that since $p_1 + p_2 = 1$, the problem is essentially reduced to studying precision of a single approximation:

$$\hat{p}_1 = f_1/t\,. \qquad (f_1, t \in \mathbb{N})$$

Thus, by setting $\hat{p}_2 = f_2/t = (t - f_1)/t$ we can see that:

$$|\delta_2| = |p_2 - f_2/t| = |1 - p_1 - (t - f_1)/t| = |p_1 - f_1/t| = |\delta_1| = \delta^*\,.$$

Hence, here we are dealing with a scalar (one-dimensional) case of Diophantine approximations, and the following result apply [3, p. 11, Theorem V]:

Fact 2. *Let α be irrational. Then there are infinitely many q and a such that*

$$q\,|q\alpha - a| < 5^{-1/2}.$$

If α is equivalent to $\frac{1}{2}\left(\sqrt{5} - 1\right)$ then the constant $5^{-1/2}$ cannot be replaced by any smaller constant. If α is not equivalent to $\frac{1}{2}\left(\sqrt{5} - 1\right)$, then there are infinitely many q and a such that:

$$q\,|q\alpha - a| < 2^{-3/2}.$$

This means, that in our case with probabilities $p_1, p_2 = 1 - p_1$, there must exist approximations for which:

$$\delta^* < \kappa \, t^{-2} \,,$$

where κ is a constant depending on p_1 in the following way (which absorbs definition of equivalence implied by above quoted result):

$$\kappa = \left[\begin{array}{ll} 5^{-1/2} \,, & \text{if } p_1 = \frac{r\psi + s}{u\psi + v}, \ \psi = \frac{\sqrt{5}-1}{2} \,, \ rv - us = \pm 1 \,, \ r, s, u, v \in \mathbb{Z} \,, \\ 2^{-3/2} \,, & \text{otherwise} \,. \end{array} \right.$$

By combining this with Lemma 1 we arrive at the second bound (9) claimed by the Theorem 2.

Our Corollaries 1, and 2 are simple consequences of Theorems 1 and 2, in which we use inequality:

$$\log_2 t \leqslant W < \log_2 t + 1.$$